

Amendments to the claims:

1. (currently amended) A method for processing a plurality of transactions within computer code being executed by a data processing system, the method comprising:

5 a) ~~examining the computer code being executed for a transaction after the execution of which a change in observable state could occur;~~

~~b) storing data for the executed computer code that are part of a sequence having a plurality of transactions between a location before the sequence and the transaction after the execution of which a change in observable state could occur, wherein the data is stored within the computer code; and~~

10 ~~c) responsive to detecting a change in observable state, committing a portion of the stored data to non-volatile memory~~
processing a sequence of transactions by, for each

transaction in the sequence:

15 storing results from execution of the instruction;

determining whether a change of observable state has
occurred;

if a change in observable state has occurred,

20 committing a portion of the stored data to non-
volatile memory.

2. (original) The method of claim 1 wherein the portion of the stored data committed includes stored data through and including the end of the stored data for the last transaction within the computer code to be fully executed.

25

3. (currently amended) The method of claim 1 further comprising:

d) responsive to detecting that execution of a transaction causes a system limitation in the data processing system to occur, committing a portion of the stored data.

30

4. (original) The method of claim 3 wherein the portion of the stored data committed includes stored data through and including the end of the stored data for the last transaction within the computer code to be fully executed.
- 5 5. (currently amended) The method of claim 1 further comprising:
 d)——maintaining a pointer to the logical beginning of the stored data and maintaining a pointer to the end of the stored data for the last transaction within the computer code to be fully executed.
- 10 6. (original) The method of claim 1 wherein the stored data are stored in a transaction buffer.
7. (original) The method of claim 6 wherein the transaction buffer comprises random access memory.
- 15 8. (original) The method of claim 1 wherein new value logging is utilized.
9. (original) The method of claim 1 wherein old value logging is utilized.
- 20 10. (original) The method of claim 1 wherein the data processing system comprises a smart card.
11. (currently amended) A data processing system configured to execute computer code having a plurality of transactions within the computer code comprising:
25 a memory;
a processor connected to the memory; and
having logic to cause the processor to process the plurality of transactions by a) ~~examining the computer code being executed for a transaction after the execution of which a change in observable state could occur;~~ b) ~~storing data for the executed computer~~
- 30

code ~~that are part of a sequence having a plurality of~~
~~transactions between a location before the sequence~~
~~and the transaction after the execution of which a~~
~~change in observable state could occur, wherein the~~
5 ~~data is stored within the computer code; and e)~~
~~responsive to detecting a change in observable state,~~
~~committing a portion of the stored data~~processing a
sequence of transactions by, for each transaction in
the sequence:

10 storing results from execution of the instruction;
determining whether a change of observable state has
occurred;

if a change in observable state has occurred,
committing a portion of the stored data to non-
15 volatile memory.

12. (original) The data processing system of claim 11 further having logic to cause
the portion of the stored data committed to include stored data through and including
the end of the stored data for the last transaction within the computer code to be fully
20 executed.

13. (currently amended) The data processing system of claim 11 further having logic
to cause the processor to process the plurality of transactions by ~~and~~ responsive to
detecting that execution of a transaction causes a system limitation in the data
25 processing system to occur, committing a portion of the stored data.

14. (original) The data processing system of claim 13 further having logic to cause
the portion of the stored data committed to include stored data through and including
the end of the stored data for the last transaction within the computer code to be fully
30 executed.

15. (original) The data processing system of claim 11 further having logic for maintaining a pointer to the logical beginning of the stored data and maintaining a pointer to the end of the stored data for the last transaction within the computer code to be fully executed.

16. (original) The data processing system of claim 11 further comprising a transaction buffer for storing the data.

17. (original) The data processing system of claim 16 wherein the transaction buffer comprises random access memory.

18. (original) The data processing system of claim 11 further having logic for utilizing new value logging.

19. (original) The data processing system of claim 11 further having logic for utilizing old value logging.

20. (original) The data processing system of claim 11 wherein the data processing system comprises a smart card.

21. (currently amended) A computer-readable medium tangibly having a program of machine-readable instructions for causing a processor to perform a method for processing a plurality of transactions within computer code being executed by a data processing system, the method comprising:

~~a) examining the computer code being executed for a transaction after the execution of which a change in observable state could occur;~~

~~b) storing data for the executed computer code that are part of the a sequence having a plurality of transactions between a location before the sequence~~

~~and the transaction after the execution of which a
change in observable state could occur, wherein the
data is stored within the computer code; and~~

5 e) ~~responsive to detecting a change in observable state,
committing a portion of the stored data~~processing a
sequence of transactions by, for each transaction in
the sequence:

storing results from execution of the instruction;

10 determining whether a change of observable state has
occurred;

if a change in observable state has occurred,
committing a portion of the stored data to non-
volatile memory.

15 22. (original) The computer-readable medium of claim 21 further having
instructions for causing the portion of the stored data committed to include stored data
through and including the end of the stored data for the last transaction within the
computer code to be fully executed.

20 23. (currently amended) The computer-readable medium of claim 21 further
having instructions for causing a processor to perform a method for processing a
plurality of transactions within computer code being executed by a data processing
system, the method comprising:

25 d) ~~responsive to detecting that execution of a transaction causes a system
limitation in the data processing system to occur, committing a portion
of the stored data.~~

24. (original) The computer-readable medium of claim 23 further having
instructions for causing the portion of the stored data committed to include stored data

through and including the end of the stored data for the last transaction within the computer code to be fully executed.

25. (currently amended) The computer-readable medium of claim 21 further
5 having instructions for causing a processor to perform a method for processing a plurality of transactions within computer code being executed by a data processing system, the method comprising:

10 ~~d)~~——maintaining a pointer to the logical beginning of the stored data and maintaining a pointer to the end of the stored data for the last transaction within the computer code to be fully executed.

26. (currently amended) The computer-readable medium of claim 21 further having instructions for causing a processor to perform a method for processing a plurality of transactions within computer code being executed by a data processing
15 system, the method comprising:

d) utilizing new value logging.

27. (original) The computer-readable medium of claim 21 further having instructions for causing a processor to perform a method for processing a plurality of
20 transactions within computer code being executed by a data processing system, the method comprising:

~~d)~~——utilizing old value logging.

28. (new) The method of claim 1, where the step committing a portion of the
25 stored data to non-volatile memory commits one or more completely processed transactions.

29. (new) The method of claim 1, where if a system limitation occurs, committing a portion of the stored data to non-volatile memory.

30

30. (new) The data processing system of claim 11, where the step committing a portion of the stored data to non-volatile memory commits one or more completely processed transactions.

5 31. (new) The data processing system of claim 11, where if a system limitation occurs, committing a portion of the stored data to non-volatile memory.

32. (new) The computer-readable medium of claim 21, where the step committing a portion of the stored data to non-volatile memory commits one or more
10 completely processed transactions.

33. (new) The computer-readable medium of claim 21, where if a system limitation occurs, committing a portion of the stored data to non-volatile memory.